REMARKS/ARGUMENTS

The non-elected claims have been canceled without prejudice or disclaimer.

Claim 26 has been amended to overcome the rejection under 35 USC §101, as discussed below.

Reconsideration of the application is respectfully requested for the following reasons:

1. Rejection of Claims 26-33 and 42 Under 35 USC §101

This rejection has been addressed by amending claims 26, from which claims 27-33 and 42 depend, to recite that the method is for protecting secret data stored in a memory of a semiconductor chip of a data carrier, and that the secret data serves as input data for one or more operations "executed on the semiconductor chip, the execution of the one or more operations causing signals detectable from outside of the data carrier." Furthermore, the recitations of the data carrier, semiconductor chip, and memory are not only included in the preamble of the claim, but are specifically tied to method steps recited in the body of the claim and that do not merely involve "insignificant post-solution activity" such as displaying results (storage of the auxiliary data in the memory occurs *before* the falsification and compensation steps rather than "post" solution, while the falsification and compensation steps are each performed on the semiconductor chip of the data carrier). As a result, the method of claim 26 is now properly tied to a machine, in compliance with the "machine or transformation" test set forth in the *Bilski* case and corresponding Examiner guidelines.

Support the amendments is found in original claim 22, which recites the data carrier, semiconductor chip, and memory, as well as detection of signals outside the semiconductor chip. Further support for the amendments is found in various descriptions in the original specification, for example in lines 1-4 on page 2 (description of the external signal pattern problem) and the paragraph bridging pages 4 and 5 (description of data carrier 1 and chip 5).

Because the method steps recited in the body of claim 26 have now been positively tied to a machine in compliance with the *Bilski* machine-or-transformation test, withdrawal of the rejection under 35 USC §101 is respectfully requested.

2. Rejection of Claims 26-33 and 42 Under 35 USC §103(a) in view of U.S. Patent Publication No. 2002/0124178 (Kocher) and U.S. Patent No. 5,655,023 (Cordery)

Reversal of the rejection under 35 USC §103(a) is respectfully requested on the grounds that the Kocher publication and the Cordery patent, whether considered individually or in any reasonable combination, fail to disclose or suggest the claimed combination of:

- falsifying secret input data by combination with auxiliary data before the execution of one or more operations;
- combining the output data determined by execution of the one or more operations with an auxiliary function value in order to compensate for the falsification of the input data; and
- the auxiliary function value having been previously determined by the execution of the one or more operations with the auxiliary data as input data in safe surroundings and stored along with the auxiliary data,

as recited in claim 26. Instead, the Kocher publication discloses computation of auxiliary data and auxiliary function values at the same time as the input data permutation computations. While the Cordery patent discloses pre-computation of secret data, it does not do so in a way that would have suggested pre-computation of secret data in the context of Kocher's real time permutation computations.

As noted previously, the Kocher publication discloses the first two steps, but not in combination with **pre-determination** in **safe surroundings** and **storage** of the auxiliary function value and auxiliary data, as claimed. To the contrary, Kocher discloses that the auxiliary data and auxiliary function value are computed **while** computing the permutation of the input data. As a result, the auxiliary function value and data are vulnerable to inspection by an attacker.

The Cordery patent, on the other hand, concerns storage of a digital key that is used to print digital tokens for a postage metering system, and has nothing to do with computation of auxiliary function values and data for use in computing input data permutations, as taught by Kocher. In fact, the function values representing encrypted data as disclosed in the Cordery patent do not correspond to, and cannot be substituted for, the function values of computed by Kocher because they are **not stored together** with auxiliary data that could be used to recover the results of performing the non-disguised operations on original input data. In other words, Cordery lacks a key feature which would have been necessary in order to successfully modify the method of Kocher to obtain the claimed invention, namely the pre-computation and storage of both auxiliary function values and auxiliary data. **Cordery teaches pre-computation and storage of one but not the other, while Kocher teaches pre-computation and storage of neither**. Even assuming that the teachings of Cordery would have been applied to the method of Kocher, and the Applicant respectfully submits that they would not have been because of the entirely different technical fields (postage payment versus protection of operations and data performed by a chip of a data carrier), the result would still not have been the claimed invention.

Furthermore, in contrast to the setting of the present invention in which the secret data to be protected by falsifying some input data are stored on a data carrier, Cordery's secret data to be protected (in the form of decryption algorithms and keys), ***is neither stored nor executed on the data carrier 104 of Cordery but rather on a secure co-processor 502 separate from the data carrier 104, and which is further protected by a tamper resistant housing 513*** (see col. 9, line 66 to col. 10, line 61 and Fig. 5 of Cordery). In other words, Cordery teaches protection of secret data by placing it in a separate tamper resistant, secure co-processor, which is completely contrary to the present invention, in which secret data on a chip is protected by falsifying operations on the chip and not by adding an additional secure, tamper resistant chip. A major purpose of the method of Kocher is to perform computations in an insecure environment, without the need to perform the operations in a secure environment or, by implication, to add a secure co-processor such as the one provided by Cordery.

Still further, merely pre-computing the auxiliary data and function value of Kocher would not by itself have resulted in the claimed invention since, in the method Kocher, the step of falsifying the input data must occur only after at least one of the one or more operations has already been executed. To carry out part of the falsifying step of Kocher before any of the operations are performed, *i.e.*, to pre-compute the auxiliary data and/or function values, would actually be contrary to the teachings of Kocher, ***since the security of the method of Kocher depends on the computation of auxiliary data and function values after the performance of at least one of the operation steps.***. According to the teachings of Kocher, the proposed modification of Kocher would make the resulting method less **secure**, and therefore not an obvious modification.

The effect of modifying the method of Kocher can be understood by considering paragraphs [0068] to [0073] of Kocher, in which the array *dataIn* corresponds to the input data, the random bits *b* correspond to the auxiliary data, and the permutations defined by the arrays *table* and *perm* arguably correspond to the one or more operations of amended claim 26. The array *perm* represents an additional permutation that is computed randomly while computing the actual permutation defined by *table*. The output data *dataOut* representing a permutation of *dataIn* according to *table* are in fact not affect by *perm* although *perm* is twice applied to the input data. As described in paragraphs [0067], [0068], the additional permutation *perm* is used to avoid processing the steps to compute the permutation *dataOut* in input order or in output order since both orders may lead to leakage of information, so that *dataOut* itself does not depend on the random array *perm*, but rather it is the order in which *dataOut*'s entries are computed that depends on the random array *perm*. The falsification of data, on the other hand, is described by adding a random bit *b* modulo 2 to the permuted input data *perm [i]*, and storing the falsified bit in an array *temp*. This is expressed in Kocher as *dataIn [p] ^ b = dataIn [perm[i]] ^ b (*where ^ is the modulo operation).

Thus, the step of falsifying data in Kocher, *i.e.*, blinding a bit of the input data by adding a random bit b modulo 2 (as can be seen in the third for-loop of the pseudo-code in paragraph [0068]), **is only performed after the step of performing the additional permutation *perm***, meaning that the corresponding auxiliary data and function value computation steps cannot be carried out before at least one of the operation steps (obtaining *perm*) is performed. This is an **essential** feature of the method of Kocher in order to prevent information leakage (paragraphs [0068] and [0069] and cannot be omitted without rendering the method of Kocher inoperative.

According to the present invention, the input data are falsified by combination with auxiliary data before the execution of the one or more operations. According to Kocher, on the other hand, the step of falsifying the input data is performed only after one of the operations, namely the permutation *perm*, has already been performed on the input data. Since Cordery's general teachings of secret data pre-computation do not provide a way to avoid Kocher's requirement that the permutation *perm* be applied to the input data before blinding of the permuted input data in order to prevent data leakage, the proposed combination could not have resulted in the claimed invention. Interchanging the permutation and falsifying (blinding) operations would be contrary to the teachings of Kocher because then the order in which the blinding steps would be executed would correspond to the standard input order (the steps would be performed in the order of index *i* from 0 to 63) with the above-mentioned risk of information leakage. Applying the permutation *perm* after the blinding steps would be useless and therefore not an obvious modification.

In Kocher, an appropriate unblinding vector is stored in the array *dataOut* and already computed together with the blinded input vector, *i.e.*, in the same for-loop defined by *dataOut ptable[p] := b*. The Examiner will note that the left hand side equals *dataOut [table[perm[i]]*, *i.e.*, that the unblinding vector is determined by applying the permutations *perm* and *table* to the random vector *b* (the random bit *b* in step *i* of the for-loop being interpreted as an entry *b[i]* of a respective vector *b*). After the permutation defined by the array *table* is applied as the second of the one or more operations, to the falsified input data in the fourth for-loop, the appropriate compensation for the prior falsification of the input data follows by means of the auxiliary function value, namely the already computed value that was stored in the array entry *dataOut [table [p]]* in the previous loop as described above. Prior to these steps, the permutation array *perm* is once more randomly permuted (in the last for-loop on page 7). This procedure ensures that the order in which the steps in the following loop are executed again is different from the previous order of steps. However, such as an arrangement is optional and only serves to further avoid information leakage. The value of *dataOut* is not affected by this procedural step.

Based on the above, it can be seen that in order to modify the method of Kocher to obtain the claimed invention, a number of changes need to be made, none of which are suggested by Cordery:

a. the ordinary artisan would have had to recognize that the method of Kocher may, at least in principle and despite explicit teachings to the contrary, be changed without any loss of security and without changing the output values by pre-computing the random bits *b and* the random permutation *perm* so that *b* and *perm* would serve as input data in addition to the actual input data *dataIn*, *dataOut*, and *table* (which would require considerable algorithmic skills not even remotely taught by Cordery);

b. the ordinary artisan would have had to further recognize that the blinding bits *b* would need to be pre-computed in safe surroundings and stored in an array of random blinding bits, for simplicity also called *b*, and that the random permutation *perm* would also have to be pre-computed in safe surroundings;

c. the ordinary artisan would have had to further recognize that the unblinding vector, stored in the vector *dataOut*, would have had to be pre-computed by applying the permutation *perm* and the permutation *table*, in that order, to the random vector *b*, *i.e.*, *dataOut [table [perm[i]] := b[i]*; and

d. the ordinary artisan would have had to provide for storing the random vector *b* representing the auxiliary data along with the unblinding vector *dataOut* representing the auxiliary function value,

with the result that the main routine to compute the actual permutation of the input array *dataIn* according to the array *table* would then comprise the following blinding steps:

```
for (i=1; i<64; i++){
        p=perm[i];                        //perm has already been pre-computed
        temp [p] := dataIn[p] ^ b[i];             //random vector b[i] has
                                                  // already been pre-computed}
```

However, even in that case, the result would not have been the claimed invention because the blinding step occurs only *after* the permutation *perm*, representing one of the one or more operations, has been applied to the input data *dataIn.* This wold be contrary to the teachings of Kocher since the execution of the permutation *perm* before blinding serves the security purpose of randomizing the order in which the blinding steps are performed. On the other hand, the pre-computed unblinding vector would then simply read *dataOut[table[i] :=b[i]* since the respective application of the permutation *perm* would also have to be omitted in order to ensure a correct un-blinding step, resulting in a contradiction that renders the propose modification of Kocher inoperative.

In conclusion, the Kocher publication specifically teaches, in paragraph [0071], lines 9-12, a method in which "*The bit order table is created n two passes, where the first assures that the table has the correct form. . .and the second introduces random order into the table"* and further that: *" Because the process of constructing the bit order table does* <u>*not*</u> *involve any secret inputs, the only security requirement for the process is that the final result be unknown to attackers*." This method is fundamentally different than that of the claimed invention, and cannot be modified to involve the claimed pre-computation without ignoring the principles explicitly taught by Kocher, while making fundamental changes to the disclosed method. Since Cordery only teaches pre-computation in the context of a secure co-processor and without storage of compensating data, Cordery does not overcome the contrary teachings of Kocher, and withdrawal of the rejection of claims 26-33 and 42 under 35 USC §103(a) is believed to be appropriate.

Having thus overcome each of the rejections made in the Official Action, expedited passage of the application to issue is requested.

Respectfully submitted,

BACON & THOMAS, PLLC

/Benjamin E. Urcia/

Date: December 4, 2009 By: BENJAMIN E. URCIA
Registration No. 33,805

BACON & THOMAS
625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314

Telephone: (703) 683-0500